Numer. Math. 70: 515–539 (1995)

Numerische Mathematik

On the convergence of a new trust region algorithm*

Ya-xiang Yuan**

Computing Center, Academia Sinica, P.O. Box 2719, Beijing 10080, China

Received March 21, 1993

Summary. In this paper we present a new trust region algorithm for general nonlinear constrained optimization problems. The algorithm is based on the L_∞ exact penalty function. Under very mild conditions, global convergence results for the algorithm are given. Local convergence properties are also studied. It is shown that the penalty parameter generated by the algorithm will be eventually not less than the l_1 norm of the Lagrange multipliers at the accumulation point. It is proved that the method is equivalent to the sequential quadratic programming method for all large k, hence superlinearly convergent results of the SQP method can be applied. Numerical results are also reported.

Mathematics Subject Classification (1991): 65K05, 49D37

1. Introduction

We consider the following nonlinear programming problem:

(1.1)
$$\min_{x\in\mathbb{R}^n} f(x)$$

(1.2)
$$\text{subject to} \quad c_i(x) = 0 \quad i = 1, 2, \ldots, m_e;$$

(1.3)
$$c_i(x) \geq 0 \quad i = m_e+1, \ldots, m$$

where $f(x)$ and $c_i(x)$ ($i = 1, \ldots, m$) are real functions defined in $\mathbb{R}^n$, and $m \geq m_e$ are two non-negative integers.

* This work was supported by a grant from Chinese National Science Foundation Grant 18901024

** Part of the work was done while the author was visiting University of Würzburg, Germany by the support of Alexander von Humboldt Foundation

Trust region algorithms for nonlinear programming problem are a class of new numerical algorithms. The first trust region method for unconstrained optimization is given by Powell [33], though the technique of trust region method is, in some sense, equivalent to that of the classical Levenberg-Marquardt method which is a method for nonlinear least squares problems and which was first given by Levenberg [26] and later re-derived by Marquardt [27]. Detailed discussions on the Levenberg-Marquardt method can be found in Moré [28]. A model trust region algorithm for nonsmooth optimization is given by Fletcher [20]. For unconstrained optimization problems and nonsmooth optimization problems, it is showed that trust region algorithms have very nice convergence properties. Convergence results can be proved provided that the approximate Hessian increases not faster than linearly. While for line search algorithms we normally have to analyze the trace and the determinant of the approximate Hessian which can be very complex (see [35] and [8]).

Trust region algorithms are generally more reliable than line search algorithms. When the approximate Hessian is ill-conditioned, a line search algorithm might give a very large search direction, which would break down the algorithm. In this case, a trust region algorithm would reduce the trust region, and in the next iteration the new trial step will be not only shorter in length but also closer to (in direction) the steepest descent direction, and hopefully the new trial step will reduce the objective function.

Trust region algorithms for unconstrained optimization have been extensively studied. Convergence results can be found in [29], [34], [38] and [40]. The trust region subproblem that needs to be solved at each iteration is studied in [23] and [30]. Recently, [31] has proposed an algorithm that combines trust region and backtracking techniques. Trust region algorithms for nonsmooth optimization are studied by [18], [20], [21], [42] and [43]. Trust region algorithms for nonlinear equations are given in [16] and [19]. For special constrained problems, trust region algorithms are analyzed by [6], [13], [14] and [24].

For constrained optimization, most trust region algorithms try to combine trust region ideas with sequential quadratical programming approach, due to the success of the SQP method. For equality constrained problems, Celis, Dennis and Tapia [11] suggested to compute trust region trial step by solving the following subproblem (the CDT subproblem):

$$\min_{\|d\|_2 \le \Delta_k} g_k^T d + \frac{1}{2} d^T B_k d \tag{1.4}$$

$$\text{subject to} \quad \|c_k + A_k^T d\|_2 \le \eta_k \tag{1.5}$$

where $g_k = g(x_k) = \nabla f(x_k)$, $c_k = c(x_k)$, $A_k = A(x_k) = \nabla c(x_k)^T$, $\Delta_k > 0$ is the trust region radius and η_k is a parameter chosen at each iteration. B_k is an $n \times n$ symmetric matrix being an approximate Hessian of the Lagrange function. In Celis, Dennis and Tapia [11] and Celis [10], η_k is chosen so that a fraction of Cauchy decrease condition on $\|c_k + A_k^T d\|_2$. The Cauchy decrease is the reduction of $\|c_k + A_k^T d\|_2$ along the steepest direction $-A_k c_k$ in the trust region, namely

(1.6) $$Decr_{\text{Cauchy}} = \|c_k\|_2 - \min_{0\le t\|A_k c_k\|_2 \le \Delta_k} \|c_k - tA_k^T A_k c_k\|_2.$$

In Powell and Yuan [39], η_k satisfies

(1.7) $$\min_{\|d\|_2\le b_2\Delta_k} \|c_k + A_k^T d\|_2 \le \eta_k \le \min_{\|d\|_2\le b_1\Delta_k} \|c_k + A_k^T d\|_2,$$

where $b_1 < b_2$ are two constants in $(0,1)$.

Another approach is based on null space and range space techniques. In these algorithms, the trial step s_k is the sum of a range space step v_k and a null space step h_k. The range space step and the null space step are also called vertical step and horizontal step respectively. The range space step v_k reduces the linearized constraint violation to zero or at least to a fraction of current constraint violation. The null space step h_k is computed by minimizing the approximate model in the null space. For equality constrained problems, Omojokun [32] obtains v_k and h_k by solving

(1.8) $$\min_{v\in\mathbb{R}^n} \|c_k + A_k^T v\|_2$$

(1.9) $$\text{subject to} \quad \|v\|_2 \le \xi\Delta_k, \quad 0<\xi<1.$$

and

(1.10) $$\min_{A_k^T h=0} g_k^T(v_k+h) + \frac{1}{2}(v_k+h)^T B_k (v_k+h)$$

(1.11) $$\text{subject to} \quad \|h\|_2 \le \sqrt{\Delta_k^2 - \|v_k\|_2^2}$$

respectively. Using an active set strategy, Omojokun extended his algorithm to general inequality constrained problems. Other algorithms that use null space step and range space step techniques are given by Byrd, Schnabel, and Shultz [9] and Vardi [41].

Recently, Dennis, El-Alem and Maciel [15] give a global convergence theory for a class of trust region algorithms for equality constrained optimization. They write the trial step s_k as the sum of s_k^t and s_k^n, where s_k^t and s_k^n are respectively the tangential and a relaxed normal components. By "relaxed", s_k^n are not required to normal to the tangent space. s_k^n is so computed such that a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints. The tangential component s_k^t is required to reduce a quadratic model of the Lagrange function in the tangent space, and this reduction is at least a fraction of Cauchy decrease of the quadratic model. El-Alem [17] and Dennis, El-Alem and Maciel [15] show that a generized Steihaug-Toint dogleg algorithm generates trial steps that satisfy the pair of fraction of Cauch decrease conditions, hence their convergence results can be applied.

Following the approach of Powell's algorithm for convex composite functions [37], Burke [4] proposes a general framework for trust region algorithms for general constrained problems, and proves the global convergence of the algorithms without regularity hypotheses. It is shown that the convergence results can be applied to a modified SQP algorithm and Fletcher's Sl_1QP algorithm. These algorithms are based on exact penalty functions of the following form:

(1.12) $$P_k(x) = f(x) + \sigma \text{dist}\{c(x)|\Gamma\},$$

where

(1.13) $$\text{dist}\{y|\Gamma\} := \inf\{||y - z||, \quad z \in \Gamma\}$$

for some fixed norm $||.||$ and

(1.14) $$\Gamma = \{c|c \in \mathbb{R}^m, c_i \geq 0, i > m_e\}.$$

Burke [4] assumes that the trial steps s_k are so computed that the predicted reduction in the exact penalty function is at least $\beta_1 \min\{\beta_2, \Delta_k\}$ for some positive constants β_1 and β_2 if x_k are bounded away from stationary points. σ_k is updated at each iteration. It is required that σ_k is not less than the norm of the Lagrange multiplier of a linearized mode at the current iterate. Thus, at each iteration (or at regular intervals) some auxiliary convex programming subproblem needs to be solved.

The algorithm we discuss in this paper is based on the L_∞ exact penalty function. Our approach is similar to that of Burke [4]. Our main contribution is that we give a simple technique for updating the penalty parameters, which does not need to solve any auxiliary subproblems. Global convergence of the algorithm is proved. It is shown that the algorithm preserves the nice local properties of the SQP method.

Though the L_∞ exact penalty is used in the paper, it can be easily seen that our results are still true if the norm $||.||_\infty$ is replaced by any other norm $||.||$. Our motivation for using the L_∞ exact penalty function is given in the following paragraph.

Most techniques for handling constraints treat all the constraints equally. This seems to be most reasonable because all constraints must be satisfied at a solution. However, the following very simple example indicates that this approach may not be the best one. Suppose we have two constraints in $\mathbb{R}^2$:

(1.15) $$c_1(x_1, x_2) = x_2 - x_1^2 = 0$$

(1.16) $$c_2(x_1, x_2) = x_1 - 1 = 0$$

which correspond to the well-known Rosenbrock function (for example, see [22]). Assume that we have a current approximate solution $(-1, 1)$. Considering the first order Taylor expansions of (1.4) and (1.5) at the point $(-1, 1)$, it is easy to see that the feasible solution for the linearized constraints is $(1, -3)$, which is not close to the actual feasible point $(1, 1)$ of (1.4)-(1.5). We can see that the point $(1, 1)$ lies in the feasible set of the linearization of the second constraint, but it is not near to any feasible point of the linearization of the first constraint. This example is too special to make a general claim because the second constraint (1.5) is exactly a linear function. However, we believe that in general the linearization of a nonlinear equation near a root may give a wrong prediction about far away roots though it can give a very good local approximation, while the linearization not near a root may be able to give a better over all prediction about all roots. Hence when there are many constraints, it is likely that the linearization of the constraints with larger residuals predict the actual feasible region better. That is the main reason for us to use the L_∞ exact penalty function in the algorithm.

Our algorithm is presented in Sect. 2. Definitions of different types of stationary points are given in Sect. 3. Global Convergence analyses of the algorithm are provided in Sect. 4 and local convergence analyses are given in Sect. 5. Numerical results are reported in Sect. 6.

2. The algorithm

We define that $c(x) = (c_1(x), \ldots, c_m(x))^T$ and $c^-(x) \in \mathbb{R}^m$ by

$$c_i^-(x) = c_i(x) \qquad i = 1, \ldots, m_e; \tag{2.1}$$

$$c_i^-(x) = \min[c_i(x), 0], \qquad i = m_e + 1, \ldots, m. \tag{2.2}$$

It is straightforward to see that the constraint conditions (1.2)-(1.3) are equivalent to the following equation:

$$||c^-(x)||_\infty = 0 \tag{2.3}$$

The L_∞ exact penalty function has the following form:

$$P_\sigma(x) = f(x) + \sigma||c^-(x)||_\infty \tag{2.4}$$

where $\sigma > 0$ is a penalty parameter. The L_∞ exact penalty is a special case of a more general exact penalty function discussed by Burke [3]. Indeed, if we define the set by (1.14) then problem (1.1)–(1.3) can be rewritten as

$$\min f(x), \qquad \text{s. t.} \quad c(x) \in \Gamma. \tag{2.5}$$

And $||c^-(x)||_\infty$ can be regarded as a distant from $c(x)$ to the set Γ: $\text{dist}(c(x), \Gamma)$.

It is obvious that a feasible point minimizing the L_∞ penalty function must be a solution of the original nonlinear programming problem (1.1)-(1.3). Under certain conditions it can be proved that a solution of the nonlinear programming problem is also a minimizer of the L_∞ penalty function.

The subproblem that our algorithm needs to solve at every iteration is based on the L_∞ exact penalty function, it has the following form:

$$\min_{d \in \mathbb{R}^n} \quad g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||(c_k + A_k^T d)^-||_\infty \tag{2.6}$$

$$\text{subject to} \quad ||d||_\infty \le \Delta_k \tag{2.7}$$

where $\sigma_k > 0$ is the penalty parameter. The superscript '-' in (2.6) has the same meaning as given in (2.1)-(2.2). Throughout this paper we use the notations

$$\phi_k(d) = g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||(c_k + A_k^T d)^-||_\infty \tag{2.8}$$

and

$$P_k(x) = P_{\sigma_k}(x) \tag{2.9}$$

Our algorithm can be stated as follows:

Algorithm 2.1.

Step 1 Given $x_1 \in \mathbb{R}^n$, $\Delta_1 > 0$ $B_1 \in \mathbb{R}^{n\times n}$ symmetric.
$\delta_1 > 0$, $\sigma_1 > 0$, $k := 1$.

Step 2 Solve subproblem (2.6)-(2.6) giving s_k;
if $s_k = 0$ then stop;

Step 3 Calculate

$$r_k = \frac{P_k(x_k) - P_k(x_k + s_k)}{\phi_k(0) - \phi_k(s_k)} \quad ; \tag{2.10}$$

if $r_k > 0$ go to Step 4;
$\Delta_{k+1} = ||s_k||_\infty/4; x_{k+1} = x_k$;
k:=k+1; go to Step 2;

Step 4 $x_{k+1} = x_k + s_k$;

$$\Delta_{k+1} = \begin{cases} \max[2\Delta_k, \quad 4||s_k||_\infty], & r_k > 0.9 , \\ \Delta_k & 0.1 \le r_k \le 0.9 , \\ \min[\Delta_k/4, \quad ||s_k||_\infty/2], & r_k < 0.1 \ ; \end{cases}$$

(2.11)

Generate B_{k+1}.

Step 5 if

$$\phi_k(0) - \phi_k(s_k) < \delta_k\sigma_k \min[\Delta_k, ||c_k^-||_\infty], \tag{2.12}$$

then $\sigma_{k+1} = 2\sigma_k$ and $\delta_{k+1} = \delta_k/4$
else $\sigma_{k+1} = \sigma_k$ and $\delta_{k+1} = \delta_k$.
$k := k + 1$ and go to Step 2.

In a practical implementation of the algorithm, the stopping criterion in Step 2 should be $||s_k||_\infty \le \epsilon$ for some small positive tolerance number ϵ instead of $s_k = 0$. The matrix B_{k+1} is normally generated by adding a lower rank matrix using an update formula which only depends on first order derivatives of the objective function and that of the constraints.

3. Stationary points

Our convergence analyses in the next section show that a cluster point of the sequence generated by our algorithm can be one of three different type points. Hence we give their definitions and show their stationarity properties.

First the following definition is standard:

Definition 3.1. x^* is called a *stationary point* if

1. $c^-(x^*) = 0$;
2. $d^Tg(x^*) \ge 0$ holds for all d satisfying

$$d^T\nabla c_i(x^*) = 0, \quad (i = 1, \ldots, m_e); \tag{3.1}$$

$$d^T\nabla c_i(x^*) \ge 0, \quad (c_i(x^*) = 0, i = m_e + 1, \ldots, m). \tag{3.2}$$

A stationary point defined above is also called a Kuhn-Tucker point. Using the Farkas lemma and part 2 of Definite 3.1, there exist $\lambda_i^*(i = 1, \ldots, m)$ such that

$$g(x^*) = \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*). \tag{3.3}$$

$$\lambda_i^* \geq 0, \quad \lambda_i^* c_i(x^*) = 0, \quad i = m_e + 1, \ldots, m. \tag{3.4}$$

It is straightforward to see that a stationary point x^* is a minimizer of the approximate problem where all the functions in (1.1)-(1.3) are replaced by their first order Taylor expansions at x^*.

Definition 3.2. x^* is called an *infeasible stationary point* if

1. $||c^-(x^*)||_\infty > 0$;
2. $\min_{d \in \mathbb{R}^n} ||(c(x^*) + A(x^*)^T d)^-||_\infty = ||c(x^*)^-||_\infty$.

By definition, an infeasible stationary point is a minimizer of the infinity norm of the linearized constraints.

Lemma 3.3. *If x^* is a infeasible stationary point as defined above, then there exist $\lambda_i^*(i = 1, \ldots, m)$ such that*

$$\sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0, \tag{3.5}$$

and

$$\lambda_i^* \geq 0 \qquad i = m_e + 1, \ldots, m. \tag{3.6}$$

Therefore the Fritz John condition

$$\lambda_0^* g(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla c_i(x^*) = 0 \tag{3.7}$$

holds if we let $\lambda_0^* = 0$.

Proof. Because x^* is an infeasible stationary point, $d = 0$ is a solution of

$$\min_{d \in \mathbb{R}^n} ||(c(x^*) + A(x^*)^T d)^-||_\infty. \tag{3.8}$$

Thus there exist $\lambda_i^*(i = 1, \ldots, m)$ such that (3.5)-(3.6) hold. Consequently (3.7) is true.

□

It is easy to see from the above lemma that the infeasible stationary point defined above is the external stationary point as defined by Burke [4, 5].

Definition 3.4. x^* is called a *singular stationary point* if

1. $c^-(x^*) = 0$;

2. there exist a sequence y_k converging to x^* such that $c(y_k)^- \neq 0$ and

$$\lim_{k\to\infty} \min_{\|d\|_\infty \le \|c(y_k)^-\|_\infty} \frac{\|(c(y_k)+A(y_k)^T d)^-\|_\infty}{\|c(y_k)^-\|_\infty} = 1 \quad (3.9)$$

It can be easily shown from the above definition that the columns of $A(x^*)$ are linearly dependent at a singular stationary point x^*. The stationary property of an singular stationary point is give as follows.

Lemma 3.5. *If x^* is a singular stationary point as defined above, then there exist $\lambda_i^*(i = 1, \ldots, m)$ such that (3.5)-(3.6) hold. Therefore x^* is also a Fritz John point.*

Proof. Assume that x^* is a singular stationary point and $y_k(k = 1, 2, \ldots)$ is a sequence that satisfy part 2 of Definition 3.4. Let $\bar{d}_k(k = 1, 2, \ldots)$ be such that

$$\|(c(y_k)+A(y_k)^T\bar{d}_k)^-\|_\infty = \min_{\|d\|_\infty \le \|c(y_k)^-\|_\infty} \|(c(y_k)+A(y_k)^T d)^-\|_\infty. \quad (3.10)$$

Using the first order necessary condition, it can be shown that there exist $\lambda_k \in \mathbb{R}^m$, $\eta_k \in \mathbb{R}^n$ and $\mu_k \ge 0$ such that

$$A(y_k)\lambda_k + \mu_k\eta_k = 0 \quad (3.11)$$

and

$$\lambda_k \in \partial\|y\|_\infty\,|_{y=(A_k^T d+c_k)^-} \quad (3.12)$$

$$\eta_k \in \partial\|d\|_\infty\,|_{d=\bar{d}_k} \quad (3.13)$$

$$\mu_k[\|\bar{d}_k\|_\infty - \|c(y_k)^-\|_\infty] = 0 \quad (3.14)$$

Thus, we have that

$$\bar{d}_k^T A(y_k)\lambda_k + \mu_k\|\bar{d}_k\|_\infty = 0 \quad (3.15)$$

which implies that

$$\mu_k = \frac{-\bar{d}_k^T A(y_k)\lambda_k}{\|c(y_k)^-\|_\infty}. \quad (3.16)$$

It can be seen from (3.12) that

$$(\lambda_k)_i \le 0, \qquad \forall\, i > m_e \quad (3.17)$$

which shows that

$$\begin{aligned} c(y_k)^T\lambda_k &= [c(y_k)^-]^T\lambda_k + [c(y_k) - c(y_k)^-]^T\lambda_k \\ &\le [c(y_k)^-]^T\lambda_k \le \|c(y_k)^-\|_\infty. \end{aligned} \quad (3.18)$$

Thus, it follows from the above inequality and (3.12) that

$$\begin{aligned} -\bar{d}_k^T A(y_k)\lambda_k &= c(y_k)\lambda_k - (c(y_k)+A(y_k)^T\bar{d}_k)^T\lambda_k \\ &= c(y_k)\lambda_k - \|(c(y_k)+A(y_k)^T\bar{d}_k)^-\|_\infty \\ &\le \|c(y_k)^-\|_\infty - \|(c(y_k)+A(y_k)^T\bar{d}_k)^-\|_\infty \end{aligned} \quad (3.19)$$

This inequality, relation (3.16) and limit (3.9) show that

$$\lim_{k \to \infty} \mu_k = 0. \tag{3.20}$$

Hence we have that

$$\lim_{k \to \infty} A(y_k)^T \lambda_k = 0 \tag{3.21}$$

Because $\|\lambda_k\|_\infty = 1$ for all k, $\{-\lambda_k\}$ has a cluster point λ^*. It follows from (3.21) and (3.17) that (3.5)-(3.7) hold. □

It should be noted that our definitions of stationary points are strongly associate with our algorithm, thus they are not standard. For example, an infeasible stationary point defined above may not be a stationary point if we use the L_1 exact penalty function instead of the L_∞ penalty function. For detailed discussions on stationary points, see Burke [2].

4. Global convergence

To study convergence properties of the algorithm, we make the following assumption:

Assumption 4.1. *1. $f(x)$ and $c_i(x)(i = 1, \ldots, m)$ are continuously differentiable.*
2. $\{x_k\}$ and $\{B_k\}$ are uniformly bounded

It is normal to assume the boundedness of $\{x_k\}$. In the case that $\{x_k\}$ is unbounded, it is very likely that the original optimization problem may be ill-posed (that is, it may have no bounded solution). In real computations, the requirement of boundedness of $\{B_k\}$ can be satisfied unless a numerical overflow happens. However, this condition can be relaxed, we shall discuss this issue later in this section.

First we can show that if the penalty parameter tends to infinity, then the infinity norm of the constraint violations converges either to zero or to a positive number.

Lemma 4.2. *If $\sigma_k \to \infty$, then $\lim_{k \to \infty} \|c_k^-\|_\infty$ exits.*

Proof. Let $k_1 < k_2 < k_3 < \ldots$ be defined such that $\sigma_{k_{i+1}} > \sigma_{k_i}$ and $\sigma_k = \sigma_{k_i}$ for all $k \in [k_i, k_{i+1})$. We define that $\bar{\sigma}_i = \sigma_{k_i}$. For any $\hat{k} > \bar{k} \to \infty$, assume $\sigma_{\bar{k}} = \bar{\sigma}_{\bar{i}}$ and $\sigma_{\hat{k}} = \bar{\sigma}_{\hat{i}}$, we have that

$$\begin{aligned} 0 \quad &\le \quad \sum_{i=\bar{k}}^{\hat{k}-1} \frac{1}{\sigma_i}[P_i(x_i) - P_i(x_{i+1})] \\ &= \quad \frac{1}{\sigma_{\bar{k}}}[f(x_{\bar{k}}) - f(x_{k_{\bar{i}+1}})] + \sum_{i=\bar{i}+1}^{\hat{i}-1} \frac{1}{\bar{\sigma}_i}[f(x_{k_i}) - f(x_{k_{i+1}})] \\ &\quad + \frac{1}{\sigma_{\hat{k}}}[f(x_{k_{\hat{i}}}) - f(x_{\hat{k}})] + \|c^-(x_{\bar{k}})\|_\infty - \|c^-(x_{\hat{k}})\|_\infty. \end{aligned} \tag{4.1}$$

Due to the boundedness of $\{x_k\}$, there exists a constant M such that $|f(x_k)| < M$ holds for all k. Thus by the definitions of σ_k and k_i, from (4.1) we have that

$$\begin{aligned}\limsup_{\hat{k}>\bar{k}\to\infty}[\|c^-(x_{\hat{k}})\|_\infty - \|c^-(x_{\bar{k}})\|_\infty] &\le \limsup_{\hat{i}>\bar{i}\to\infty} 2M\sum_{i=\bar{i}}^{\hat{i}}\frac{1}{\bar{\sigma}_i}\\ &\le \limsup_{\bar{i}\to\infty} 4M/\bar{\sigma}_{\bar{i}}\\ &= 0 .\end{aligned} \tag{4.2}$$

Relation (4.1) also implies that

$$\limsup_{k\to\infty}\|c^-(x_k)\|_\infty < \infty . \tag{4.3}$$

The non-negativeness of $\|c^-(x_k)\|_\infty$ gives that

$$\liminf_{k\to\infty}\|c^-(x_k)\|_\infty \ge 0 . \tag{4.4}$$

From (4.2)-(4.4), it is easy to prove that $\lim_{k\to\infty}\|c^-(x_k)\|_\infty$ must exist. □

Lemma 4.3. *If* $\lim_{k\to\infty}\sigma_k = \infty$ *and* $\lim_{k\to\infty}\|c^-(x_k)\|_\infty > 0$, *then the sequence* $\{x_k\}$ *is not bounded away from infeasible stationary points.*

Proof. Assume the lemma were not true, there would exist an integer k_0 and a compact set Ω such that $x_k \in \Omega$ for all $k > k_0$, $c^-(x) \neq 0$ for all $x \in \Omega$, and there exists no infeasible stationary point in Ω. By the definition of infeasible stationary points and the closedness of Ω, there exists a constant $\mu > 0$ such that

$$\min_{\|d\|_\infty\le 1}\|(c(x)+A(x)^T d)^-\|_\infty \le \|c^-(x)\|_\infty - \mu \tag{4.5}$$

holds for all $x \in \Omega$. We define the vector $d(x)$ to be a solution for

$$\|(c(x)+A(x)^T d(x))^-\|_\infty = \min_{\|d\|_\infty\le 1}\|(c(x)+A(x)^T d)^-\|_\infty . \tag{4.6}$$

Hence, using the convexity of $\|(c(x)+A(x)^T d)^-\|_\infty$, the boundedness of $\{B_k\}$ and the definition of s_k, we can prove that

$$\begin{aligned}\phi_k(0)-\phi_k(s_k) &\ge \phi_k(0)-\phi_k(d(x_k)\min[1, \frac{\Delta_k}{\|d(x_k)\|_\infty}])\\ &\ge \sigma_k\mu\min[1, \frac{\Delta_k}{\|d(x_k)\|_\infty}] + O(\Delta_k)\\ &\ge \bar{\mu}\sigma_k\Delta_k \ge \bar{\mu}\sigma_k\min[\Delta_k, \|c_k^-\|_\infty]\end{aligned} \tag{4.7}$$

for all large k, $\bar{\mu}$ being a positive constant. Because $\sigma_k \to \infty$, we have that $\delta_k \to 0$ and that (2.12) holds for infinitely many k. This contradicts (4.7). The contradiction verifies that the lemma is true. □

Similarly, we can prove the following lemma:

Lemma 4.4. *If* $\lim_{k\to\infty} \sigma_k = \infty$ *and* $\lim_{k\to\infty} \|c^-(x_k)\|_\infty = 0$, *then the sequence* $\{x_k\}$ *is not bounded away from singular stationary points.*

One direct corollary of Lemmas 3.6 and 3.7 is that $\{\sigma_k\}$ remains bounded if the optimization problem has neither infeasible stationary point nor singular stationary point.

Now we study the case when σ_k does not tend to infinity. Our main global convergence theorem states as follows:

Theorem 4.5. *If* $\sigma_k = \sigma$ *for all large* k, *then the sequence* $\{x_k\}$ *is not bounded away from* $K-T$ *points.*

Proof. Without loss of generality, we can assume that $\sigma_k = \sigma$, $\delta_k = \delta$ and

$$\phi_k(0) - \phi(s_k) > \delta\sigma \min[\Delta_k, \|c_k^-\|_\infty] \tag{4.8}$$

hold for all k.

Let $\bar{\Omega}$ be the set of all accumulation points $\bar{x}$ of $\{x_k\}$ such that $c^-(\bar{x}) = 0$. If the theorem is not true, for any $\bar{x} \in \bar{\Omega}$, there exists a constant $\bar{\eta} > 0$ such that

$$\min_{\|d\|_\infty \le 1} \bar{\phi}(d) \le \bar{\phi}(0) - \bar{\eta}, \tag{4.9}$$

where

$$\bar{\phi}(d) = g(\bar{x})^T d + \frac{1}{2} M \|d\|_2^2 + \sigma \|(c(\bar{x}) + A(\bar{x})^T d)^-\|_\infty \tag{4.10}$$

and M is a positive constant that satisfies $\|B_k\|_2 \le M$ for all k. Because $\bar{\Omega}$ is a compact set, it is easy to show that there exist positive constants $\hat{\mu}$ and $\hat{\eta}$ such that

$$\min_{\|d\|_\infty \le \Delta_k} \phi_k(d) \le \phi_k(0) - \hat{\eta} \min\{\Delta_k, 1\} \tag{4.11}$$

provided that $\text{dist}(x_k, \bar{\Omega}) \le \hat{\mu}$, where $\text{dist}(x, Y) = \min\{\|x - y\|_2, y \in Y\}$. From the definition of $\bar{\Omega}$, there exists a constant $\hat{\delta} > 0$ such that $\|c^-(x_k)\|_\infty \ge \hat{\delta}$ if $\text{dist}(x_k, \bar{\Omega}) \ge \hat{\mu}$. Hence from (4.8), we have that

$$\begin{aligned} \phi_k(0) - \phi_k(s_k) &\ge \delta\sigma \min\{\Delta_k, \|c_k^-\|_\infty\} \\ &\ge \delta\sigma \min\{\Delta_k, \hat{\delta}\} \end{aligned} \tag{4.12}$$

whenever $\text{dist}(x_k, \bar{\Omega}) \ge \hat{\mu}$. From (4.11), (4.12) and the boundedness of Δ_k, there exists a positive constant $\bar{\delta}$ such that

$$\phi_k(0) - \phi_k(s_k) \ge \bar{\delta}\Delta_k \tag{4.13}$$

holds for all k.

We call an iteration is a "*good iteration*" if $r_k \ge 0.1$. We denote the set of good iterations by K, it follows from (4.13) that

$$
(4.14) \qquad \begin{aligned} \infty &> \sum_{k=1}^{\infty}[P_k(x_k) - P_k(x_{k+1})] \\ &\geq \sum_{k\in K} 0.1\bar{\delta}\Delta_k. \end{aligned}
$$

The fact that $\sum_{k\in K}\Delta_k < \infty$ implies that $\sum_{k=1}^{\infty}\Delta_k < \infty$ (see [34]). Thus $\Delta_k \to 0$. From our continuity assumptions, we have that

$$
(4.15) \qquad P_k(x_k) - P_k(x_k + s_k) = \phi_k(0) - \phi_k(s_k) + o(\Delta_k).
$$

The above relation and (4.13) implies that $r_k \to 1$, which shows that $\Delta_{k+1} \geq \Delta_k$ for all sufficiently large k. This contradicts that $\sum_{k=1}^{\infty}\Delta_k < \infty$. The contradiction shows that our theorem is true. □

It should be noted that our global convergence results do not require the assumption of the linearly independence of the gradients of the active constraints.

Now we consider the relaxation for the boundedness of $\{B_k\}$. In a practical algorithm B_k is usually updated by certain updating techniques. Hence generally it is not easy to prove explicitly that $\{B_k\}$ is uniformly bounded. However, for some update formulae, it is not difficult to establish that

$$
(4.16) \qquad ||B_k||_2 \leq \alpha k
$$

for a positive constant α(for example, see [34] and [38]). The above inequality is obviously satisfied if the change in B_k at every iteration is uniformly bounded. More specifically, (4.16) is automatically satisfied if $||B_{k+1}|| \leq ||B_k|| + \hat{\alpha}$ holds for every k. Following the techniques of Powell [38] and Yuan [43], we can establish the convergence results provided that (4.16) holds for all k. For the rest of this section, we give up the assumption that $\{B_k\}$ is uniformly bounded. Instead, we only assume that B_k is updated from iteration to iteration such that (4.16) is satisfied.

We modify our algorithm slightly, replacing (2.12) by

$$
(4.17) \qquad \phi_k(0) - \phi_k(s_k) < \delta_k\sigma_k \min\{\Delta_k, 1/k \ \ ||c_k^-||_\infty\}.
$$

It is easy to see that the validity of Lemma 4.2 is independent of $||B_k||$ and independent of inequality (2.12). Hence it is obvious that Lemma 4.2 remains true. As for Lemma 4.3, if it does not hold, similar to (4.7) it can be shown that

$$
(4.18) \qquad \phi_k(0) - \phi_k(s_k) \geq \bar{\mu}\sigma_k \min\{\Delta_k, 1/k\},
$$

which would give a contradiction. Therefore, Lemma 4.3 still holds for the modified algorithm under the condition (4.16). In a similar way, we can re-establish Lemma 4.4.

In the case that σ_k is bounded, if Theorem (4.5) is not true, similar to (4.13) we can show that

$$
(4.19) \qquad \phi_k(0) - \phi_k(s_k) \geq \bar{\delta}\min\{\Delta_k, 1/k\}.
$$

which implies that

$$\sum_{k \in K} \min\{\Delta_k, 1/k\} < \infty. \tag{4.20}$$

For any subsequence that satisfies that $k\|s_k\|_\infty \to 0$ we have that $\|s_k\|_2\|B_k\|_2 \to 0$ which implies that

$$P_k(x_k) - P_k(x_k + s_k) = \phi_k(0) - \phi_k(s_k) + o(\|s_k\|_\infty). \tag{4.21}$$

(4.18) and relation $k\|s_k\|_\infty \to 0$ give that

$$\phi_k(0) - \phi_k(s_k) \geq \bar{\mu}\sigma_k\|s_k\|_\infty, \tag{4.22}$$

for sufficiently large k. It follows from (4.21) and (4.22) that $r_k \to 1$. Thus $k\|s_k\|_\infty$ is bounded away from zero for all k such that $r_k \leq 0.1$. Therefore there exists a positive constant β such that $\|s_k\|_\infty \geq \beta/k$ for all $k \notin K$ where K is the set of good iterations as defined in the proof of Theorem 4.5. The relation $\|s_k\|_\infty \geq \beta/k$ for all $k \notin K$ implies that $\Delta_k \geq \hat{\beta}/k$ for some positive constant $\hat{\beta}$ and all k (see the proof of Lemma 4.2 in [31]). Consequently (4.20) gives that

$$\sum_{k \in K} 1/k < \infty. \tag{4.23}$$

The convergence results are proved under the assumption that the trial step s_k is an exact solution of the subproblem (2.6)-(2.7). However, by modifying the proofs, one can show that the convergence results remain valid provided the predicted reduction of the penalty function satisfies

$$Pred_k = \phi_k(0) - \phi_k(s_k) \geq \delta\epsilon_k \min\{\Delta_k \ \epsilon/\|B_k\|\} \tag{4.24}$$

for some positive constant δ, where ϵ is the violation of the KT conditions which is defined by

$$\epsilon = \|c_k^-\| + \|g_k - A_k\lambda_k\| \tag{4.25}$$

and λ_k being an approximate multiplier at the current point x_k and it satisfies that $(\lambda_k)_i \geq 0$, $i > m_e$. The condition (4.24) is an extension of that in Powell's trust region algorithm for unconstrained optimization [34]. It can be shown that condition (4.24) allows many inexact solutions of subproblem (2.6)-(2.7). We believe that s_k can be easily computed to satisfy (4.24) even when B_k is indefinite. This is very important for trust region algorithms, as allowing indefinite approximation Hessian matrices is an important property and also a motivation of trust region algorithms.

5. Local convergence results

Throughout this section, we make the following assumption:

Assumption 5.1. *1. $f(x)$ and $c_i(x)(i = 1, \ldots, m)$ are continuously differentiable;*
2. $x_k \to x^*$;
3. $\sigma_k = \sigma^*$ for all large k;
4. $\{B_k\}$ is bounded.

It should be noted that our global convergence analyses in the previous section do not imply that x_k will always converge to a point. Let x^* be an accumulation point of $\{x_k\}$, results in the previous section imply that x^* is one of the three different types of stationary defined in Sect. 3. Part 3 of the above assumption implies that x^* is a $K-T$ point, therefore $c^-(x^*) = 0$ and $d^T g(x^*) \geq 0$ holds for all d satisfying

$$d^T \nabla c_i(x^*) = 0, \qquad (i = 1, \ldots, m_e); \tag{5.1}$$

$$d^T \nabla c_i(x^*) \geq 0, \qquad (c_i(x^*) = 0, i = m_e + 1, \ldots, m). \tag{5.2}$$

Let λ^* be the Lagrange multiplier at x^*. If second order sufficient condition holds at x^*, for any given $\sigma > \|\lambda^*\|_1$, there exist a positive constant δ and a function $\eta(t) > 0$ defined in $(0, \delta)$ such that

$$P_\sigma(x) \geq P_\sigma(x^*) + \eta(t) \tag{5.3}$$

holds for all x satisfying

$$t \leq \|x - x^*\| \leq \delta, \tag{5.4}$$

for all $t \in (0, \delta)$. Thus, if $\sigma_k = \sigma^* > \|\lambda^*\|_1$ for all large k (which is very likely true due to the following lemma), and if $s_k \to 0$, it can be seen from the monotonic decreasing property of $P_k(x_k)$ and relation (5.3) that x_k converges to x^*.

First we have the following lemma:

Lemma 5.2. *If Assumption 5.1 is satisfied, then x^* is a $K-T$ point of problem (1.1)-(1.3), and*

$$\sigma^* \geq \min\{\|\lambda\|_1 : \lambda \in \Omega(x^*)\} \tag{5.5}$$

where $\Omega(x^*)$ is the set of all Lagrange multipliers, namely $\lambda \in \Omega(x^*)$ if and only if

$$g(x^*) - A(x^*)\lambda = 0 \tag{5.6}$$

and

$$\lambda_i c_i(x^*) = 0 \tag{5.7}$$

$$\lambda_i \geq 0 \tag{5.8}$$

hold for all $i = m_e + 1, \ldots, m$.

Proof. It is obvious that x^* is a $K-T$ point. Hence we only need to prove inequality (5.5). Because s_k is the minimum of problem (2.6)-(2.7), there exist $\lambda_k \in \mathbb{R}^m$, $\eta_k \in \mathbb{R}^n$ and $\mu_k \geq 0$ such that

$$g_k + B_k s_k + A_k \lambda_k + \mu_k \eta_k = 0 \tag{5.9}$$

where

$$\lambda_k/\sigma_k \in \partial\|y\|_\infty |_{y=(A_k^T d + c_k)^-} \tag{5.10}$$

$$\eta_k \in \partial\|d\|_\infty |_{d=s_k} \tag{5.11}$$

and $\mu_k(\Delta_k - \|s_k\|_\infty) = 0$. It is easy to see that $\|\eta_k\|_1 = 1$ and $\|\lambda_k\|_1 \leq \sigma_k$. If

$$\liminf_{k\to\infty} \mu_k = 0, \tag{5.12}$$

then it follows that

$$\liminf_{k\to\infty} \|g_k + B_k s_k + A_k \lambda_k\| = 0. \tag{5.13}$$

The above relation implies that there exist a $\bar{\lambda} \in \Omega(x^*)$ and a subsequence of $\{-\lambda_k\}$ that converges to $\bar{\lambda}$. Consequently it can be shown that

$$\sigma^* \geq \limsup_{k\to\infty} \|\lambda_k\|_1 \geq \|\bar{\lambda}\|_1 \tag{5.14}$$

which implies (5.5). Now assume that (5.12) does not hold. There exists a positive constant $\hat{\delta}$ such that

$$\mu_k \geq 2\hat{\delta} \tag{5.15}$$

for all large k. The above inequality implies that $\|s_k\|_\infty = \Delta_k$. Define the set $K^* = \{k \mid \quad r_k > 0\}$. We have that

$$\begin{aligned} \lim_{k\to\infty, k\in K^*} \Delta_k &= \lim_{k\to\infty, k\in K^*} \|s_k\|_\infty \\ &= \lim_{k\to\infty, k\in K^*} \|x_{k+1} - x_k\|_\infty = 0. \end{aligned} \tag{5.16}$$

Therefore, it follows that

$$\limsup_{k\to\infty} \Delta_k \leq \limsup_{k\to\infty\ k\in K^*} 2\Delta_k = 0. \tag{5.17}$$

Thus, it follows from the above relation, (5.15) and the boundedness of $\{\|B_k\|\}$ that

$$\begin{aligned} \phi_k(0) - \phi_k(s_k) &\geq -s_k^T g_k - \frac{1}{2} s_k^T B_k s_k - \sum_{i=1}^m (\lambda_k)_i s_k^T \nabla c_i(x_k) \\ &= \mu_k \|s_k\|_\infty + \frac{1}{2} s_k^T B_k s_k \\ &\geq \hat{\delta}\|s_k\|_\infty = \hat{\delta}\Delta_k \end{aligned} \tag{5.18}$$

for all large k. The above inequality, (5.17) and our continuity assumptions on $f(x)$ and $c(x)$ imply that

$$\lim_{k\to\infty} r_k = 1 \tag{5.19}$$

which indicates that $\Delta_{k+1} \geq \Delta_k$ for all sufficiently large k. This contradicts (5.17). Hence our lemma is true. □

To continue our local analyses we need the following assumptions:

Assumption 5.3. *1. $\nabla c_i(x^*)(i \in E \cup I^*)$ are linearly independent, where $E = \{1, 2, \ldots, m_e\}$ and $I^* = \{i \mid c_i(x^*) = 0, m_e < i \leq m\}$;*

2. $\sigma^* > \|\lambda^*\|_1$ where λ^* is the unique Lagrange multipliers at the solution x^*.

Define the matrix $\bar{A}^* = \{a_i(x^*)\}(i \in E \cup I^*)$. Due to 1) of Assumption 5.3, $\bar{A}^*$ has full column rank. Hence for any vector $\lambda \in \mathbb{R}^{|E\cup I^*|}$ we have that

$$\|\bar{A}^*\lambda\|_2 \geq \|\lambda\|_2/\|(\bar{A}^*)^+\|_2. \tag{5.20}$$

We also define $\bar{K}$ be the set of iterations in which the trial step makes the linearized constraints zero, namely

$$\bar{K} = \{k \mid (c_k + A_k^T s_k)^- = 0\}. \tag{5.21}$$

Lemma 5.4. *If the conditions in Assumptions 5.1 and 5.3 are satisfied, then there exists a positive constant $\hat{\delta}$ such that*

$$\phi_k(0) - \phi_k(s_k) \geq \hat{\delta}\Delta_k \tag{5.22}$$

$$\|s_k\|_\infty = \Delta_k \tag{5.23}$$

$$x_{k+1} = x_k + s_k \tag{5.24}$$

holds for all large $k \notin \bar{K}$, and

$$\lim_{k\to\infty, k\notin\bar{K}} r_k = 1. \tag{5.25}$$

Proof. From (5.9), the following relation

$$\begin{aligned} \mu_k &:= \|g_k - A_k\lambda^* + B_k s_k + A_k(\lambda^* - \lambda_k)\|_1 \\ &= \|A(x^*)(\lambda^* - \lambda_k)\|_1 + O(\|x_k - x^*\|_\infty + \|s_k\|_\infty). \end{aligned} \tag{5.26}$$

holds for all large k. If $k \notin \bar{K}$, $\|\lambda_k\|_1 = \sigma_k$. Hence, it follows from (5.26) and (5.20) that

$$\begin{aligned} \mu_k &= \|A(x^*)(\lambda^* - \lambda_k)\|_1 + o(1) \\ &\geq \|A(x^*)(\lambda^* - \lambda_k)\|_2 + o(1) \\ &\geq \|\lambda^* - \lambda_k\|_2/\|(\bar{A}^*)^+\|_2 + o(1) \\ &\geq \frac{\|\lambda^* - \lambda_k\|_1}{\sqrt{n}\|(\bar{A}^*)^+\|_2} + o(1) \\ &\geq \frac{\|\lambda_k\|_1 - \|\lambda^*\|_1}{\sqrt{n}\|(\bar{A}^*)^+\|_2} + o(1) \\ &= \frac{\sigma^* - \|\lambda^*\|_1}{\sqrt{n}\|(\bar{A}^*)^+\|_2} + o(1) \geq \frac{\sigma^* - \|\lambda^*\|_1}{2\sqrt{n}\|(\bar{A}^*)^+\|_2} \end{aligned} \tag{5.27}$$

holds for all large $k \notin \bar{K}$. The above inequality implies that $\|s_k\|_\infty = \Delta_k$. Therefore (5.23) is true for all large $k \notin \bar{K}$. Now similar to (5.18), from (5.27) we can show that there exists a positive constant $\hat{\delta}$ such that (5.22) holds for all $k \notin \bar{K}$. (5.22) implies (5.25). And (5.24) follows from (5.25). □

Lemma 5.5. *If the conditions in Assumptions 5.1 and 5.3 are satisfied, then*

(5.28) $$(c(x_k) + A(x_k)^T s_k)^- = 0$$

holds for all large k.

Proof. If $\bar{K}$ defined by (5.21) is a finite set then from Lemma 5.4 (5.22)-(5.25) holds for all large k. (5.25) and (5.22) implies that

(5.29) $$\sum_{k=1}^{\infty} \Delta_k < \infty.$$

On the other hand, it follows from (5.25) that

(5.30) $$\Delta_{k+1} \geq \Delta_k$$

which contradicts (5.29). Therefore we have shown that $\bar{K}$ has infinitely many elements.

If the lemma is not true, there exist infinitely many $k \notin \bar{K}$. Hence there exist a subsequence $k_i (i = 1, 2, \dots)$ such that $k_i - 1 \in \bar{K}$ and $k_i \notin \bar{K}$ for all i. If $x_{k_i - 1} = x_{k_i}$ and i is sufficiently large, then we have that

(5.31) $$\begin{aligned} \phi_{k_i-1}(0) - \phi_{k_i-1}(s_{k_i-1}) &\geq \phi_{k_i}(0) - \phi_{k_i}(s_{k_i}) \\ &\geq \hat{\delta}\Delta_{k_i} \geq \hat{\delta}\|s_{k_i-1}\|_\infty/4. \end{aligned}$$

The above inequality implies that $r_{k_i-1} \to 1$, which contradicts $x_{k_i-1} = x_{k_i}$. Therefore

(5.32) $$x_{k_i} = x_{k_i-1} + s_{k_i-1}$$

for all large i. The above relation and $k_i - 1 \in \bar{K}$ gives that

(5.33) $$\begin{aligned} c_{k_i}^- &= [c_{k_i-1} + A_{k_i-1}^T s_{k_i-1} + O(\|s_{k_i-1}\|^2)]^- \\ &= O(\|s_{k_i-1}\|^2) = O(\Delta_{k_i}^2) \end{aligned}$$

Thus, by the definition of $\phi_k(d)$ and (5.33) we have that

(5.34) $$\phi_{k_i}(0) - \phi_{k_i}(s_{k_i}) \leq -g_{k_i}^T s_{k_i} + O(\Delta_{k_i}^2)$$

(5.22) and (5.34) imply that $g_{k_i}^T s_{k_i} \leq -\hat{\delta}\|s_{k_i}\|_\infty + O(\|s_{k_i}\|_\infty^2)$. Without loss of generality, we assume that $s_{k_i}/\|s_{k_i}\|_\infty \to s^*$. It is obvious that $(g^*)^T s^* < 0$. Due to $g^* - A^*\lambda^* = 0$, we have that

(5.35) $$-(s^*)^T A(x^*)\lambda^* = -(g^*)^T s^*$$

which implies that

$$\|(A(x^*)^T s^*)^-\|_\infty \ge -(g^*)^T s^* / \|\lambda^*\|_1. \tag{5.36}$$

Now we show that

$$c_{k_i}^T \lambda^* = o(\Delta_{k_i}). \tag{5.37}$$

If the above relation is not true, due to (5.33), there exists positive constant $\bar{\delta}$ such that

$$c_{k_i}^T \lambda^* \ge \bar{\delta} \Delta_{k_i}. \tag{5.38}$$

Hence $\|\lambda_{k_i} - \lambda^*\|$ will be bounded away from zero for all large i. Consequently, due to the uniqueness of λ^*, for all large i it follows that

$$\phi_{k_i-1}(0) - \phi_{k_i-1}(s_{k_i-1}) \ge \bar{\bar{\delta}} \Delta_{k_i-1} \tag{5.39}$$

for some constant $\bar{\bar{\delta}} > 0$. This implies that

$$\|s_{k_i-1}\|_1 = \Delta_{k_i-1}, \qquad \Delta_{k_i} \ge 2\Delta_{k_i-1} \tag{5.40}$$

for all large i. Because k_i is a subsequence and because Δ_k can not always increase, without loss of generality, we can assume that t_i be an index such that $j \in \bar{K}$ for all $t_i \le j < k_i$,

$$\lambda_{t_i} \to \lambda^*, \tag{5.41}$$

and

$$\|s_j\|_1 = \Delta_j, \qquad \Delta_{j+1} = 2\Delta_j \tag{5.42}$$

for all $t_i < j < k_i$. (5.41) and $t_i \in \bar{K}$ indicate that

$$\begin{aligned} c_{t_i+1}^T \lambda^* &= O(\|s_{t_i}\|^2) = O(\Delta_{t_i+1}^2) \\ &= O(\Delta_{k_i}^2). \end{aligned} \tag{5.43}$$

It follows from (5.38) and (5.43) that

$$c_{k_i}^T \lambda^* - c_{t_i+1}^T \lambda^* \ge \frac{1}{2} \bar{\delta} \Delta_{k_i}. \tag{5.44}$$

(5.42) implies that $\|x_{k_i} - x_{t_i+1}\| = O(\Delta_{k_i})$. Thus, Using $g^* = A^* \lambda^*$ and the above relation, we have that

$$f(x_{k_i}) - f(x_{t_i+1}) \ge \frac{1}{4} \bar{\delta} \Delta_{k_i} \tag{5.45}$$

for all large i. Because (5.33) and

$$c_{t_i+1}^- = o(\Delta_{k_i}), \tag{5.46}$$

it follows from (5.45) that

$$P_{k_i}(x_{k_i}) - P_{t_i+1}(x_{t_i+1}) > 0 \tag{5.47}$$

for sufficiently large i. This is a contradiction. Therefore (5.37) is true. From (5.37), we have that

$$\begin{aligned} \|(c_{k_i}+A_{k_i}^{\mathrm{T}}s_{k_i})^-\|_\infty\|\lambda^*\|_1 &\geq -[(c_{k_i}+A_{k_i}^{\mathrm{T}}s_{k_i})^-]^{\mathrm{T}}\lambda^* \\ &\geq -(c_{k_i}+A_{k_i}^{\mathrm{T}}s_{k_i})^{\mathrm{T}}\lambda^* \\ &= -s_{k_i}^{\mathrm{T}}A_{k_i}\lambda^* - c_{k_i}^{\mathrm{T}}\lambda^* \\ &= -s_{k_i}^{\mathrm{T}}g^* + o(\Delta_{k_i}) \\ &= -(g^*)^{\mathrm{T}}s^*\|s_{k_i}\|_\infty + o(\|s_{k_i}\|_\infty). \end{aligned} \tag{5.48}$$

Thus, it follows that

$$\|(c_{k_i}+A_{k_i}^{\mathrm{T}}s_{k_i})^-\|_\infty \geq -(g^*)^{\mathrm{T}}s^*\|s_{k_i}\|_\infty/\|\lambda^*\|_1 + o(\|s_{k_i}\|_\infty). \tag{5.49}$$

Therefore, it follows from the definition of $\phi_k(d)$, (5.33) and (5.49) that

$$\begin{aligned} \phi_{k_i}(0)-\phi_{k_i}(s_{k_i}) &= -g_{k_i}^{\mathrm{T}}s_{k_i} - \sigma^*\|(c_{k_i}+A_{k_i}^{\mathrm{T}}s_{k_i})^-\|_\infty + O(\|s_{k_i}\|_\infty^2) \\ &\leq -(g^*)^{\mathrm{T}}s^*\|s_{k_i}\|_\infty + \sigma^*(g^*)^{\mathrm{T}}s^*\|s_{k_i}\|_\infty/\|\lambda^*\|_1 + o(\|s_{k_i}\|_\infty) \\ &= \left([\frac{\sigma^*}{\|\lambda^*\|_\infty}-1](g^*)^{\mathrm{T}}s^* + o(1)\right)\|s_{k_i}\|_\infty < 0 \end{aligned} \tag{5.50}$$

for all large i, which contradicts the definition of s_{k_i}. The contradiction shows that our lemma is true. □

The above lemma shows that for sufficiently large k, the trial step s_k computed is a solution of the following problem:

$$\min_{d\in\mathbb{R}^n} g_k^{\mathrm{T}}d + \frac{1}{2}d^{\mathrm{T}}B_kd \tag{5.51}$$

subject to

$$c_i(x_k)+d^{\mathrm{T}}\nabla c_i(x_k) = 0 \qquad i=1,2,\dots,m_e \tag{5.52}$$

$$c_i(x_k)+d^{\mathrm{T}}\nabla c_i(x_k) \geq 0, \qquad i=m_e+1,\dots,m \tag{5.53}$$

$$\|d\|_\infty \leq \Delta_k. \tag{5.54}$$

Thus if k is very large and the trust region bound is inactive the trial step generated by Algorithm 2.1 are the same as search directions of the sequential quadratic programming method (see [7]).

To prove local superlinear convergence of algorithms for nonlinear constrained optimization, one normally needs the following second order sufficient condition and a good approximation of the Hessian of Lagrange function in the null space of the gradients of active constraints.

Assumption 5.6. *1. $f(x)$ and $c_i(x)(i=1,\dots,m)$ are twice continuously differentiable;*

2. the following inequality

$$d^{\mathrm{T}}W^*d > 0 \tag{5.55}$$

holds for all nonzero d that satisfy

(5.56) $$d^T\nabla c_i(x^*) = 0; \quad i = 1, \ldots, m_e$$

(5.57) $$d^T\nabla c_i(x^*) \geq 0; \quad i \in I^*,$$

where

(5.58) $$W^* = \nabla^2 f(x^*) - \sum_{i=1}^{m}(\lambda^*)_i \nabla^2 c_i(x^*);$$

3.

(5.59) $$\lim_{k\to\infty} \frac{\|\bar{P}(B_k - W^*)s_k\|_2}{\|s_k\|_2} = 0$$

where $\bar{P}$ is a projection from $\mathbb{R}^n$ to the null space of $(\bar{A}^*)^T$.

From superlinearly convergence results of the SQP method (for example, see [1]), we have the following lemma:

Lemma 5.7. *If the conditions of Assumptions 5.1, 5.3 and 5.6 are satisfied, if $\|s_k\|_\infty < \Delta_k$ for all large k, then we have that*

(5.60) $$\lim_{k\to\infty} \frac{\|x_k + s_k - x^*\|_\infty}{\|x_k - x^*\|_\infty} = 0.$$

Unfortunately, the result (5.60) is not exactly the same as the Q-superlinear convergence of the iterate points $\{x_k\}$:

(5.61) $$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|_\infty}{\|x_k - x^*\|_\infty} = 0,$$

since (5.60) can not guarantee $x_{k+1} = x_k + s_k$ for all large k. In fact, Algorithm 2.1 is eventually a trust region algorithm for minimizing the nonsmooth function $P_{\sigma^*}(x)$ due to the fact that $\sigma_k = \sigma^*$ for all large k. Unfortunately, nonsmoothness of the objective function may cause unnecessary reduction of the trust region bound and lead to only linear convergence (for example, see [42]).

The second order step technique for nonsmooth optimization is proposed by [21], and its Q-superlinear convergence result is proved in [44]. In our case, the second order correction subproblem is

(5.62) $$\min_{d\in\mathbb{R}^n} \; g_k^T(s_k + d) + \frac{1}{2}(s_k + d)^T B_k(s_k + d) + \sigma_k\|(c(x_k + s_k) + A_k^T d)^-\|_\infty$$

(5.63) $$s.t. \quad \|s_k + d\|_\infty \leq \Delta_k.$$

Using the trust region subproblems (2.6)-(2.7) and (5.62)-(5.63), we can construct the following second order correction step trust region algorithm, which is a modification of Algorithm 2.1.

Algorithm 5.8

Step 0. Given $x_1 \in \mathbb{R}^n$, $\Delta_1 > 0$, $B_1 \in \mathbb{R}^{n\times n}$ symmetric,
$\delta_1 > 0$, $\sigma_1 > 0$, $\epsilon > 0$, $k := 1$.

Step 1. Solve (2.6)-(2.7) giving s_k;
if $Pred_k \le \epsilon$ and $\|c_k^-\|_\infty \le \epsilon$ then stop;
if $\|c_k^-\|_\infty - \|(c_k + A_k s_k)^-\|_\infty < \epsilon$ and $\|(c_k + A_k s_k)^-\|_\infty > \epsilon$
then $\sigma_k := 11\sigma_k$ and $\delta_k := \delta_k/11$;
if (2.12) holds then $\sigma_{k+1} = 2\sigma_k$ and $\delta_{k+1} = \delta_k/4$
else $\sigma_{k+1} = \sigma_k$ and $\delta_{k+1} = \delta_k$.

Step 2. Calculate r_k;
if $r_k > 0.75$ go to Step 5;
solve (5.62)-(5.63) giving $\hat{s}_k$;
compute approximate ratio:

$$\bar{r}_k = r_k + \frac{\bar{\phi}_k(0) - \bar{\phi}_k(\hat{s}_k)}{Pred_k}; \tag{5.64}$$

where $\bar{\phi}_k(d)$ is the objective function in (5.62);
if $r_k < 0.25$ go to Step 3;
if $\bar{r}_k \in [0.9, 1.1]$ then set $\Delta_{k+1} = 2\Delta_k$ else set $\Delta_{k+1} = \Delta_k$;
go to Step 6.

Step 3. If $\bar{r}_k < 0.75$ go to Step 4;
calculate f and c at $x_k + s_k + \hat{s}_k$;
if $P_k(x_k + s_k + \hat{s}_k) \ge P_k(x_k + s_k)$ then go to Step 4;
calculate $\hat{r}_k = \frac{P_k(x_k) - P_k(x_k + s_k + \hat{s}_k)}{Pred_k}$;
in computation set $s_k := s_k + \hat{s}_k$ and $r_k := \hat{r}_k$;
if $r_k \ge 0.75$ go to Step 5;
if $r_k \ge 0.25$ go to Step 6.

Step 4. $\Delta_{k+1} = \|s_k\|_\infty/2$; go to Step 6.

Step 5. If $\|s_k\|_\infty < \Delta_k$ then $\Delta_{k+1} := \Delta_k$ and go to Step 6;
if $r_k > 0.9$ then $\Delta_{k+1} := 4\Delta_k$ else $\Delta_{k+1} := 2\Delta_k$.

Step 6. If $r_k > 0$ go to Step 7;
$x_{k+1} = x_k$; $B_{k+1} = B_k$; $k := k + 1$ and go to Step 1.

Step 7. Compute ∇f and ∇c at $x_k + s_k$;
generate B_{k+1};
$x_{k+1} = x_k + s_k$; $k := k + 1$ and go to Step 1.

For more details of second order correction step technique and convergence analyses, see [21] and [44]. Similar to the analyses in [44], the following result can be proved:

Theorem 5.9. *If the conditions of Assumptions 5.1, 5.3 and 5.6 are satisfied, then Algorithm 5.8 with $\epsilon = 0$ either terminates at a $K - T$ point or generates a sequence $\{x_k\}$ that converges Q-superlinearly:*

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|_\infty}{\|x_k - x^*\|_\infty} = 0. \tag{5.65}$$

Moreover, the trust region bound Δ_k is bounded away from zero, and trust region constraint $||d||_\infty \le \Delta_k$ is inactive for all large k.

6. Numerical results

A FORTRAN subroutine is programmed to test Algorithm 5.8. To solve the nonsmooth subproblem (2.6)-(2.7), we rewrite it into the following equivalent quadratic program:

$$\min_{\bar{d}\in\mathbb{R}^{n+1}} \bar{g}_k^{\mathrm{T}}\bar{d} + \frac{1}{2}\bar{d}^{\mathrm{T}}\bar{B}_k\bar{d} \tag{6.1}$$

subject to

$$d_{n+1} + (c_i(x_k) + d^{\mathrm{T}}\nabla c_i(x_k)) \ge 0 \qquad i = 1, \ldots, m \tag{6.2}$$

$$d_{n+1} - (c_i(x_k) + d^{\mathrm{T}}\nabla c_i(x_k)) \ge 0 \qquad i = 1, \ldots, m_e \tag{6.3}$$

$$-\Delta_k \le d_i \le \Delta_k \qquad i = 1, \ldots, n; \tag{6.4}$$

$$d_{n+1} \ge 0 \tag{6.5}$$

where $d = (d_1, \ldots, d_n)^{\mathrm{T}}$, $\bar{d} = (d^{\mathrm{T}}\ d_{n+1})^{\mathrm{T}}$, $\bar{g}^{\mathrm{T}} = (g^{\mathrm{T}}, \sigma)$, and $\bar{B}_k$ is the $(n+1)\times(n+1)$ matrix that is defined by

$$\bar{B}_k = \begin{bmatrix} B_k & 0 \\ 0 & 0 \end{bmatrix}. \tag{6.6}$$

Similarly, the second order correction subproblem (5.62)-(5.63) can also be rewritten as a quadratic programming problem. We solve the quadratic programming subproblems by Fletcher's Harwell subroutine VE02AD.

The test examples that we have run are from [25]. For each problem, we choose initial parameters $\Delta_1 = 10$, $\sigma_1 = 10$, and $\delta_1 = 0.01$. The error tolerance is $\epsilon = 10^{-10}$. The algorithm is also terminated when the infinite norm of the residual for Kuhn-Tucker condition

$$\kappa_k = ||c_k||_\infty + ||g_k - A_k\lambda_k||_\infty \tag{6.7}$$

is less than 10^{-10}, where λ_k is the Lagrange multipliers for the subproblem (2.6)-(2.7). For comparison, the test examples are also solved by Powell's subroutine VMCWD, which is a very successful line search algorithm. For more details about VMCWD, see [12] and [36]. The error tolerance for VMCWD is also set to 10^{-10}, which means that VMCWD is terminated when the quadratic programming model indicates that the objective function plus suitably weighted multiples of the constraint functions are predicted to differ from their optimal values by at most 10^{-10}.

For both VMCWD and our algorithm, B_1 is set to I and B_{k+1} is updated by the Powell's safeguarded BFGS update formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^{\mathrm{T}} B_k}{s_k^{\mathrm{T}} B_k s_k} + \frac{\eta_k \eta_k^{\mathrm{T}}}{\eta_k^{\mathrm{T}} s_k} \tag{6.8}$$

Table 1.

			VMCWD		Our algorithm	
Problem	n	m	NI-NF-NG	Residual	NI-NF-NG	Residual
HS6	2	1	7-9-9	1.06×10^{-5}	7-12-8	1.18×10^{-11}
HS28	3	1	3-5-5	2.09×10^{-15}	10-10-9	3.84×10^{-7}
HS34	3	8	8-9-9	4.44×10^{-16}	7-13-8	9.62×10^{-11}
HS61	3	2	Failed		9-9-7	8.56×10^{-8}
HS71	4	10	5-6-6	8.34×10^{-9}	6-6-6	5.09×10^{-8}
HS80	5	13	8-10-10	2.20×10^{-12}	9-11-8	1.84×10^{-9}
HS93	6	8	13-16-16	5.22×10^{-7}	20-26-14	2.13×10^{-5}
HS100	7	4	19-27-27	2.82×10^{-7}	39-45-21	4.26×10^{-6}
HS113	10	8	13-18-18	1.39×10^{-7}	19-21-15	4.12×10^{-8}
HS119	16	40	8-13-13	1.06×10^{-6}	17-18-13	6.08×10^{-7}

where η_k is the convex combination of y_k and $B_k s_k$ such that η_k is as close to y_k as possible and that $\eta_k^T s_k > 0.1 s_k^T B_k s_k$ is satisfied. y_k is the difference of the gradients of Lagrange function at $x_k + s_k$ and x_k, namely

$$(6.9) \qquad y_k = g(x_k + s_k) - g(x_k) - \sum_{i=1}^{m} (\lambda_k)_i [\nabla c_i(x_k + s_k) - \nabla c_i(x_k)].$$

VMCWD updates B_k at every iteration. In our algorithm, gradients of the objective function and the constraint functions are only calculated at acceptable points, therefore y_k is only available at successful iterations (namely either the trial step s_k or the second order correction step $\hat{s}_k$ is accepted.). Thus, in our algorithm B_k is updated only at successful iterations.

The calculations were done by a DEC 2100 workstation in double precision arithmetic. Both algorithms solved all the test problems that we run, except that VMCWD failed to solve problem HS61, where the gradients of constraint functions are linearly dependent at the initial point. The numerical results are listed in Table 1. In the table, the problems are numbered in the same way as in [25]. For example, "HS6" means problem 6 in Hock and Schittkowski (1981). NI, NF and NG means numbers of iterations, function evaluations and gradient evaluations respectively. "Residual" is the infinity norm of the residual for the $K - T$ condition (6.7) at the computed solution.

The numerical results in Table 1 favor VMCWD slightly. The numbers of gradient evaluations of both algorithms are about the same. But, unfortunately, our algorithm requires slightly more function evaluations. Our limited numerical tests indicate that our algorithm is comparable to VMCWD.

Because VMCWD, which uses watch-dog techniques, performs better than the original VF02AD, we believe that it is interesting to investigate the possibilities of combining trust region and watch-dog techniques.

To solve the trust region subproblem (2.6)-(2.7) efficiently is important to the efficiency of our algorithm. Our current implementation of the algorithm uses VE02AD to solve trust region subproblems. When problems HS61, HS71 and HS80 were solved by our algorithm, in some iterations VE02AD failed to provide accurate solutions for subproblem (6.1)-(6.5). It would be desirable to have a nice subroutine to solve the trust region subproblem (2.6)-(2.7) directly.

Acknowledgements. This work was done when I was visiting University of Würzburg. I am very grateful to Professor Dr. J. Stoer for providing excellent facilities. I would like to thank Professor M.J.D. Powell for giving me the VMCWD subroutine, thank Dr. I Duff and Dr. N. Gould for the Harwell subroutine VE02AD, and thank Professor K. Schittkowski for testproblem subroutines. I would like to thank an anonymous referee and associate editor Jorge Moré for their helpful suggestions and very careful reading of this paper, which improve the paper greatly.

References

1. Boggs, P.T., Tolle, J.W., Wang P. (1982) On the local convergence of quasi-Newton methods for constrained optimization. SIAM J. Control Opt. 20, 161–171
2. Burke, J.V. (1989): A sequential quadratic programming method for potentially infeasible mathematical programs. J. Math. Anal. Appl. 139, 319–351
3. Burke, J.V. (1991): An exact penalization viewpoint of constrained optimization SIAM J. Control and Opt. 29, 968–998
4. Burke, J.V. (1992): A robust trust region method for constrained nonlinear programming problems. SIAM J. Optimization 2, 325–347
5. Burke, J.V., Moré, J.J. (1988): On the identification of active constraints. SIAM J. Numer Anal 25, 1197–1211
6. Burke, J.V., Moré, J.J., Toraldo, G. (1990): Convergence properties of trust region methods for linear and convex constraints. Math. Prog. 47, 305–336
7. Burke, J.V., Han, S.P. (1989): A robust sequential quadratic programming method. Math. Prog. 43, 277–303
8. Byrd, R., Nocedal, J. (1989): A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. SIAM J. Numer. Anal. 26, 727–732
9. Byrd, R., Schnabel, R.B., Shultz, G.A. (1987): A trust region algorithm for nonlinearly constrained optimization. SIAM Journal of Numerical Analysis 24, 1152–1170
10. Celis, M.R.: A Trust Region Strategy for Nonlinear Equality Constrained Optimization Ph. D. thesis, Dep. of Math. Rice University, USA
11. Celis, M.R., Dennis, J.R., Tapia, R.A. in: Boggs, P.T., Byrd, R.H., Schnabel R.B. (eds.): Numerical Optimization. SIAM, Philadelphia, 1985, pp. 71–82
12. Chamberlain, R.M., Lemarechal, C., Pedersen, H.C., Powell, M.J.D. (1982): The watchdog techniques for forcing convergence in algorithms for constrained optimization. Math. Prog. Study 16, 1–17
13. Conn, A.R., Gould, N.I.M., Toint, Ph.L. (1988): Global convergence of a class of trust region algorithms for optimization with simple bounds. SIAM J. Numer. Anal. 25, 433–460
14. Conn, A.R., Gould, N., Toint, Ph.L. (1992): Convergence properties of minimization algorithms for convex constraints using a structured trust region. Technical Report 92/11, Facultés Universitaires de Namur Belgium
15. Dennis, J.E. Jr., El-Alem, M., Maciel, M.C.: A global convergence theory for general trust region-based algorithms for equality constrained optimization. TR 92-28 Department of Computational and Applied Mathematics, Rice University, Houston, USA
16. Duff, I., Nocedal, J., Reid, J. (1987): The use of linear programming for the solution of sparse sets of nonlinear equations. SIAM J. Sci. Stat. Comput. 8, 99–108
17. El-Alem, M (1988): A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization. Ph. D. Thesis, Dept of Mathematical Sciences, Rice University, USA
18. El Hallabi, M.: A global convergence theory for a class of trust region methods for nonsmooth optimization. Report MASC TR 90-16, Rice University, USA
19. El Hallabi, M., Tapia, R.: A global convergence theory for arbitrary norm trust-region methods for nonlinear equations. Report MASC TR 87-25, Rice University, Houston, USA
20. Fletcher, R. (1982): A model algorithm for composite NDO problem. Math. Prog. Study 17, 67–76

21. Fletcher, R. (1982): Second order correction for nondifferentiable optimization. In: Watson, G.A. (ed.): Numerical Analysis, Springer, Berlin, pp. 85–115
22. Fletcher, R. (1987): Practical Methods of Optimization (second edition) John Wiley and Sons Chichester
23. D.M. Gay, Computing optimal local constrained step, SIAM J. Sci. Stat. Comp. 2(1981) 186-197.
24. Gay, D.M. (1978): A trust region approach to linearly constrained optimization. In: Griffiths D.F. (ed.): Lect. Notes Math. 1066: Numerical Analysis pp. 72–105, Springer, Berlin,
25. Hock, W., Schittkowski, K. (1981): Test Examples for Nonlinear Programming Codes. Lect Notes Economics and Mathematical Systems, vol. 187. Springer, Berlin
26. Levenberg, K. (1944): A method for the solution of certain nonlinear problems in least squares. Quart. Appl. Math. **2**, 164–166
27. Marquardt, D.W. (1963): An algorithm for least-squares estimation of nonlinear inequalities. SIAM J. Appl. Math. **11**, 431–441
28. Moré, J.J. (1978): The Levenberg-Marquardt algorithm. implementation and theory. In: Watson, G.A. (ed.): Proc. Dundee Conference on Numerical Analysis, Springer, Berlin
29. Moré, J.J. (1983): Recent developments in algorithms and software for trust region methods. In: Bachem, A., Grötschel, M., Korte, B. (eds.): Mathematical Programming: The State of the Art, pp. 258–287, Springer, Berlin
30. Moré, J.J., Sorenson, D.C. (1983): Computing a trust region step. SIAM J. Sci. Stat. Comp. 4, 553–572
31. Nocedal, J., Yuan, Y. (1991): Combining trust region and line search techniques. Report NAM06 EECS, Northwestern University, USA
32. Omojokun, E.O. (1989): Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints. Ph. D. Thesis, University of Colorado at Boulder
33. Powell, M.J.D. (1970): A new algorithm for unconstrained optimization. In: Rosen, J.B., Mangassarian, O.L., Ritter, K. (eds.): Nonlinear Programming pp 31–66, Academic Press, New York
34. Powell, M.J.D. (1975): Convergence properties of a class of minimization algorithms. In: Mangassarian, O.L., Meyer, R.R., Robinson, S.M. (eds.): Nonlinear Programming 2, pp. 1–27, Academic Press New York
35. Powell, M.J.D. (1976): Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In: Cottle, R.W., Lemke, C.E. (eds.): Nonlinear Programming SIAM-AMS Proc. 9, pp. 53–72, SIAM, Philadephia
36. Powell, M.J.D. (1982): VMCWD: A FORTRAN subroutine for constrained optimization DAMTP Report 1982/NA4, University of Cambridge, England
37. Powell, M.J.D. (1983): General algorithm for discrete nonlinear approximation calculations. In: Chui, C.K., Schumaker, L.L., Ward, L.D. (eds.): Approximation Theory IV, pp. 187–218, Academic Press, N.Y.
38. Powell, M.J.D. (1984): On the global convergence of trust region algorithm for unconstrained optimization. Math. Prog. **29**, 297–303
39. Powell, M.J.D., Yuan, Y. (1991): A trust region algorithm for equality constrained optimization. Math. Prog. **49**, 189–211
40. Sorensen, D.C. (1982): Newton's method with a model trust region modifications. SIAM J. Numer. Anal. **19**, 409–426
41. Vardi, A. (1985): A trust region algorithm for equality constrained minimization: convergence properties and implementation. SIAM J. Numer. Anal. **22**, 575–591
42. Yuan, Y. (1984): An example of only linearly convergence of trust region algorithms for nonsmooth optimization. IMA J. Numer. Anal. **4**, 327–335
43. Yuan, Y. (1985): Conditions for convergence of trust region algorithms for nonsmooth optimization. Math. Prog. **31**, 220–228
44. Yuan, Y. (1985): On the superlinear convergence of a trust region algorithm for nonsmooth optimization. Math. Prog. **31**, 269–285

This article was processed by the author using the LaTeX style file *pljour1m* from Springer-Verlag.